

Claims

What is claimed is:

1. An application development system, comprising:
a development tool;
a software component; and
a type descriptor adapted to access metadata associated with the software component, the type descriptor operative to dynamically provide information associated with the software component to the development tool to facilitate application development.
2. The system of claim 1, wherein the type descriptor is further adapted to store the information associated with the software component.
3. The system of claim 1, wherein the information associated with the software component comprises at least one of types, members, attributes, properties and events.
4. The system of claim 1, wherein the software component is adapted to provide a custom type descriptor interface.
5. The system of claim 4, wherein the custom type descriptor interface is adapted to manipulate information associated with the software component and to provide the manipulated information to the type descriptor.
6. The system of claim 5, wherein the information provided to the development tool is one of the metadata and the manipulated information received from the custom type descriptor interface.
7. The system claim 1, wherein the type descriptor provides the information associated with the software component in response to a request from a developer.

8. An application development system, comprising:
a development tool; and,
a software component adapted to provide a custom type descriptor interface which is adapted to provide information associated with the software component to a type descriptor, the type descriptor adapted to access metadata associated with the software component wherein the type descriptor is further adapted to receive information associated with the software component from the custom type descriptor interface and to provide one of the metadata or information received from the custom type descriptor interface to the development tool to facilitate application development.
9. The system of claim 8, wherein the custom type descriptor interface is adapted to manipulate the information associated with the software component before providing the information to the type descriptor.
10. The system of claim 8, wherein the type descriptor is further adapted to store one of the metadata or information received from the custom type descriptor interface.
11. The system of claim 8, wherein the metadata comprises at least one of types, members, attributes, properties and events.
12. The system of claim 8, wherein the information provided by the custom type descriptor interface comprises at least one of types, members, attributes, properties and events.
13. The system claim 8, where the type descriptor provides the information associated with the software component in response to a request from a developer.
14. An application development system, comprising:
a plurality of contained components;

a development tool adapted to provide a container adapted to facilitate communication among the plurality of contained components and further adapted to provide a site comprising a plurality of type descriptor filter service interfaces for manipulating information associated with the plurality of contained components; and

a type descriptor adapted to access metadata associated with the plurality of contained components wherein the type descriptor is further adapted to receive information associated with the plurality of contained components from the plurality of type descriptor filter service interfaces and to provide one of the metadata or information receiving from the plurality of type descriptor filter service interfaces to the development tool to facilitate application development.

15. The system of claim 14 wherein at least one of the contained components is adapted to provide a custom type descriptor interface.

16. In a component based environment, a method for developing an application comprising:

determining whether an instance of a component implements a custom type interface; invoking the custom type descriptor interface of instance of the component, the custom type descriptor interface manipulating information regarding the instance of the component;

receiving information regarding the instance of the component from the custom type descriptor interface; and,

storing the information regarding the instance of the component.

17. The method of claim 16, further comprising at least one of the following acts:

receiving a request from a development tool for information regarding the instance of the component;

discovering metadata associated with the instance of the component; and,

reporting the information regarding the instance of the component to the development tool.

18. In a component based environment, a method for developing an application comprising:

receiving information regarding an instance of a component;
determining whether the instance of the component is contained by a container;
determining whether any other contained component desires to modify information regarding the instance of the component;
modifying the information regarding the instance of the component;
determining whether the container implements a type descriptor filter service interface for the instance of the component;
manipulating the information regarding the instance of the component by the type descriptor filter service interface; and,
storing the information regarding the instance of the component.

19. The method of claim 18, further comprising at least one of the following acts:

receiving a request from a development tool for information regarding the instance of the component;
discovering metadata associated with the instance of the component;
determining whether the instance of the component implements a custom type interface;
invoking the custom type descriptor interface of instance of the component, the custom type descriptor interface manipulating information regarding the instance of the component;
manipulating information regarding the instance of the component;
receiving information regarding the instance of the component from the custom type descriptor interface; and,
reporting the information regarding the instance of the component to the development tool.

20. A computer-readable medium having computer-executable instructions for executing at least a portion of the method of claim 16.

21. An application development system comprising:

means for determining whether an instance of a component implements a custom type descriptor interface;

means for invoking the custom type descriptor interface of instance of the component wherein the custom type descriptor interface is adapted to manipulate information regarding the instance of the component;

means for manipulating information regarding the instance of the component;

means for receiving information regarding the instance of the component from the custom type descriptor interface; and,

means for storing the information regarding the instance of the component.

22. The application development system of claim 21, further comprising at least one of means for receiving a request from a development tool for information regarding the instance of the component, means for discovering metadata associated with the instance of the component and means for reporting the information regarding the instance of the component to the development tool.